

Mis à jour le 05/02/2025

S'inscrire

Formation Flutter & Dart

3 jours (21 heures)

Présentation

Notre Formation [Flutter](#) et [Dart](#) vous permettra de créer des interfaces utilisateur, d'ajouter des fonctionnalités et de tester et corriger les bugs plus rapidement. Flutter est un framework créé par Google, pour la création d'interfaces natives de haute qualité sur iOS et Android en un temps record. Dart est, lui, un langage de programmation généraliste développé utilisé pour créer des applications Web, serveur et mobiles, ainsi que pour les appareils de l'Internet des objets (IoT).

Dans cette formation, vous allez maîtriser le Framework Flutter et le langage Dart. Avec une introduction aux outils de développement, nous aborderons le langage Dart dans son ensemble, et passerons sur la programmation Orientée Objet. Nous apprendrons ensuite le langage Flutter, du Hello World à la base de donnée, en passant par les widgets.

La formation utilisera les dernières versions stables en date du projet ([Flutter 3.27](#) et [Dart 3.6](#) à ce jour).

Objectifs

- Être capable de concevoir des Interfaces Utilisateurs
- Être capable d'interagir avec des apis REST
- Être capable de stocker des données dans le cache
- Maîtriser des concepts de programmation asynchrone
- Être capable de gérer les états de l'application
- Pouvoir créer des Interfaces responsives
- Être capable d'accéder aux fonctionnalités native des dispositifs

Public visé

- Développeur
- Lead Developer

Pré-requis

- Connaissance dans un langage de programmation orienté objet (Java, C#...)
- [Tester Mes Connaissances](#)

PRÉ-REQUIS LOGICIELS

- Les dernières mises à jour de l'OS (Mac ou Windows)
- Flutter SDK mis à jour Android Studio avec le SDK Android installé (pour les tests sur simulateur Android)
- Visual Studio Code
- [Flutter Live Preview Plugin](#) pour Visual Studio Code

RECOMMANDATIONS DE LECTURES AVANT LA FORMATION

- Nous vous recommandons le [site personnel de Dane Mackier](#) qui fournit une tonne de code snippets clé en main ainsi que des tutoriels vidéos complètement qui démontrent qu'il n'y a pas de limites avec Flutter !
- Le [blog de Christian Findlay](#) développeur australien qui met à disposition des articles techniques sur Dart, les composants Flutter, les Tests, l'immutabilité et l'loC, la programmation asynchrone (async/await)
- [L'accessibilité](#) (Flutter team en parle dans son blog)
- L'injection de dépendance et autres trucs et astuces de Flutter dans le [blog de Reso Coder](#)

Programme de notre formation Flutter & Dart

Jour 1 : Introduction à Flutter et Dart

- Présentation de Flutter et de Dart
- Configuration de l'environnement de développement sur macOS et Windows
- Utilisation de l'éditeur de code recommandé par Flutter
- Utiliser Flutter Version Management (FVM) en local
- Bases de la programmation Dart : variables, types de données, conditions, boucles, fonctions
- Concepts de widgets et d'interfaces utilisateur
- Exploration des widgets intégrés et création de vos propres widgets
- Astuces de débogage

Le langage Dart

- Introduction à Dart

- La syntaxe
 - Types de données
 - Structures de contrôle
 - Fonctions
 - Null-Safety
- Concepts de POO en Dart
 - Types optionnels
 - Programmation asynchrone
 - Isolates
 - Streams
 - Mixins
 - Génériques
 - Méthodes d'extension
 - Loop For-in

Création d'une belle interface utilisateur avec Flutter

- Utilisation du fichier Pubspec.yaml pour incorporer des dépendances, des ressources personnalisées et des polices de caractère
- Introduction à la méthode build() des widgets et utilisation de widgets de disposition tels que Columns, Rows, Containers et Cards
- Utilisation de MediaQuery pour créer une interface utilisateur réactive en fonction de la taille de l'écran
- Exécution de l'application Flutter avec le widget MaterialApp et la méthode runApp()
- Compréhension de la philosophie de Flutter de l'interface utilisateur en tant que code et refactoring de widgets

Jour 2 : Gestion d'état

- Gestion d'état dans les applications Flutter
- Utilisation de Stateful et Stateless widgets
- Compréhension des méthodes du cycle de vie des widgets Stateful
- Compréhension de la façon de passer des données aux objets State via le widget Stateful
- Gestion d'état avancée
- Utilisation du widget TextField pour prendre des entrées utilisateur
- Mise en place du Provider pour partager les données entre les écrans
- Utilisation de la classe ChangeNotifier pour la gestion d'état
- Création de la classe Model pour stocker les données
- Utilisation de la classe Consumer pour consommer les données
- Exemples de gestion d'état avec Provider

Navigation

- Introduction à la navigation entre les écrans
- Utilisation de la classe Navigator pour naviguer entre les écrans
- Création de différentes vues
- Ajout de boutons pour naviguer entre les écrans

- Compréhension de la façon de passer des données en arrière à l'aide du widget Navigator
- Création de tabs, side drawers et navigation basée sur les piles

Animations

- Utilisation de l'animation de transition pour une expérience utilisateur plus fluide
- Utilisation de la classe Hero pour créer des animations de transition personnalisées
- Introduction aux animations avancées de transition
- Ajout d'animations et de transitions de page
- Création d'animations personnalisées pour une expérience utilisateur plus riche

Jour 3 : Création d'applications Crossplatform et connectées

- Comprendre Flutter sur toutes les plateformes (Web, Mobile, Desktop)
- Adapter l'application Flutter pour les différentes plateformes
- Les notifications Push : comment envoyer des notifications à vos utilisateurs
- L'utilisation des fonctionnalités natives des appareils mobiles, comme la caméra
- Utilisation de packages tiers pour ajouter des fonctionnalités supplémentaires à l'application
- Récupération des données de localisation sur iOS et Android

API Request & Manipuler du JSON

- Apprentissage de la programmation asynchrone en Dart avec async/await et l'API
- Compréhension des Futures
- Manipulation de contenu JSON
- Envoi de requêtes HTTP pour récupérer des données à partir d'un backend
- Gestion des exceptions en Dart avec try/catch et throw

Firestore

- Présentation de Firestore
- Configuration d'un projet
- Utilisation de Firestore Authentication pour authentifier les utilisateurs dans l'application Flutter
- Utilisation de Firestore Cloud Firestore pour stocker et récupérer des données structurées dans l'application Flutter
- Améliorer sa compréhension de l'asynchronisme avec les Streams
- Création d'une application de messages instantanés

Module complémentaire (+1 jour) : Tests et Intégration Continue

Tests automatisés

- Vue d'ensemble des tests de Flutter
- Écriture de tests unitaires et de tests de widgets
- Utilisation de tests d'intégration pour les tests de bout en bout

Génération de code avec Freezed

- Vue d'ensemble de la génération de code avec Freezed
- Génération de modèles et de classes d'états immuables
- Amélioration de la qualité du code et réduction du code en double

CI/CD avec CodeMagic ou Github Actions

- Vue d'ensemble des pipelines et des workflows CI/CD
- Configuration de CI/CD pour les applications Flutter avec CodeMagic ou Github Actions
- Automatisation des processus de construction, de test et de déploiement

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.